

Retrieving Data Using http:

1. Add http library in your index.html file, this file already included with rxJs library so no need to include that.

index.html

```
<!DOCTYPE html>
<html>

<head>
  <title>CAP Product Management</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="node_modules/bootstrap/dist/css/bootstrap.css" rel="stylesheet"
/>
  <link href="app/app.component.css" rel="stylesheet" />

  <!-- 1. Load libraries -->
  <!-- IE required polyfills, in this exact order -->
  <script
src="node_modules/angular2/es6/dev/src/testing/shims_for_IE.js"></script>
  <script src="node_modules/es6-shim/es6-shim.min.js"></script>
  <script src="node_modules/systemjs/dist/system-polyfills.js"></script>

  <script src="node_modules/angular2/bundles/angular2-polyfills.js"></script>
  <script src="node_modules/systemjs/dist/system.src.js"></script>
  <script src="node_modules/rxjs/bundles/Rx.js"></script>
  <script src="node_modules/angular2/bundles/angular2.dev.js"></script>

  <!-- required for http -->
  <script src="node_modules/angular2/bundles/http.dev.js"></script>

  <!-- 2. Configure SystemJS -->
  <script>
    System.config({
      packages: {
        app: {
          format: 'register',
          defaultExtension: 'js'
        }
      }
    });
    System.import('app/main')
      .then(null, console.error.bind(console));
  </script>
</head>
```

```
<body>
  <pm-app>Loading App.</pm-app>

</body>

</html>
```

2. Include HTTP_PROVIDERS service in your app.component.ts file as mentioned below:

app.component.ts

```
import { Component } from 'angular2/core';
import { ProductListComponent } from '../products/product-list.component';
import { ProductService } from '../products/product.service';
import { HTTP_PROVIDERS } from 'angular2/http';
import 'rxjs/Rx'; //Load all features

@Component({
  selector: 'pm-app',
  template: `
    <div>
      <h1>{{pageTitle}}</h1>
      <pm-products></pm-products>
    </div>
  `,
  directives: [ProductListComponent],
  providers: [ProductService, HTTP_PROVIDERS]
})

export class AppComponent{
  pageTitle: string= 'CAP Product Management';
}
```

3. Sending an Http Request

- a. Now you can remove hard coded product list from product.service.ts file. And replace that with http request.

product.service.ts

```
import {Injectable} from 'angular2/core';
import {IProduct} from '../product';
import {Http, Response} from 'angular2/http';
import {Observable} from 'rxjs/Observable';

@Injectable()
```

```

export class ProductService{
    private _productUrl = 'api/products/products.json';

    constructor(private _http : Http){

    }

    getProducts(): Observable<IProduct[]>{
        return this._http.get(this._productUrl)
            .map((response:Response) => <IProduct[]>response.json())
            .do( data => console.log("All: " +JSON.stringify(data)))
            .catch(this.handleError);
    }

    private handleError(error:Response){
        console.error(error);
        return Observable.throw(error.json().error || 'Server Error');
    }
}

```

4. Subscribe your product in your product-list.component.ts file as mentioned below.

```

import { Component, OnInit } from 'angular2/core';
import { IProduct } from './product';
import { ProductFilterPipe } from './product-filter.pipe';
import { StarComponent } from '../shared/star.component';
import { ProductService } from './product.service';
@Component({
    selector: 'pm-products',
    templateUrl: 'app/products/product-list.component.html',
    styleUrls: ['app/products/product-list.component.css'],
    pipes: [ProductFilterPipe],
    directives: [StarComponent]
})
export class ProductListComponent implements OnInit{
    pageTitle: string = 'Product List';
    imageWidth: number = 50;
    imageMargin: number = 2;
    showImage: boolean = false;
    listFilter: string ;
    products: IProduct[];
    errorMessage: string;
}

```

```

    constructor(private _productService: ProductService){

    }

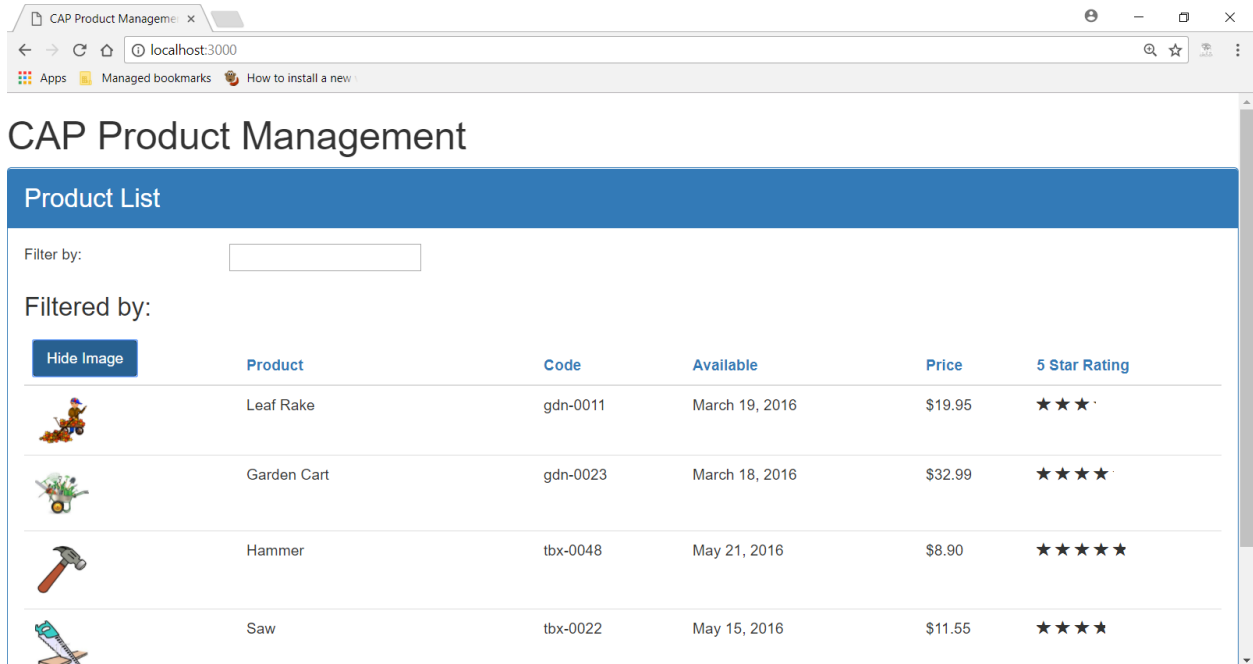
    toggleImage(): void{
        this.showImage = !this.showImage;
    }

    ngOnInit(): void{
        this._productService.getProducts()
            .subscribe(
                products => this.products= products,
                error => this.errorMessage = <any>error);
    }

    onRatingClicked(message: string): void{
        this.pageTitle = 'Product List: ' + message;
    }
}

```





5. Check whether products.json file has been include in your api/products directory.
Now you can see the below output



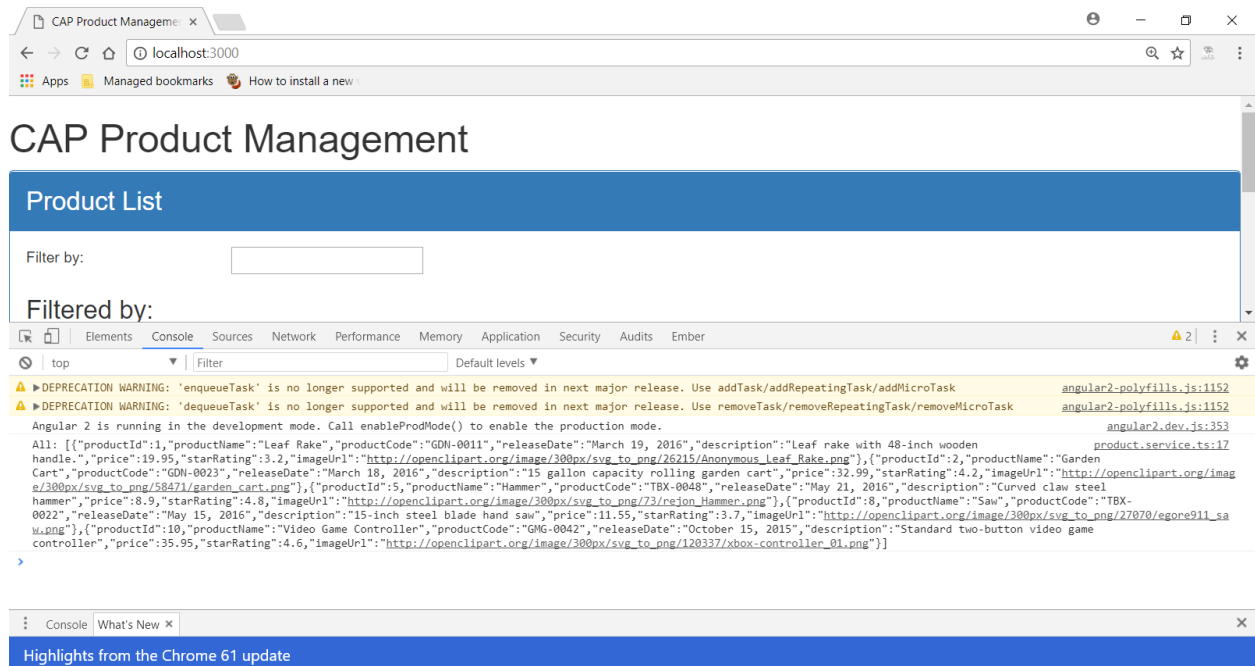
Product List

Filter by:

Filtered by:

Hide Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn-0011	March 19, 2016	\$19.95	★★★★
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	★★★★★
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx-0022	May 15, 2016	\$11.55	★★★★★

Also check your console window in the browser:



Navigation and Routing

6. To setup routers you should start from index.html file. In index.html file include router library.

```
<!DOCTYPE html>
<html>

<head>
  <base href="/" />
  <title>CAP Product Management</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link href="node_modules/bootstrap/dist/css/bootstrap.css" rel="stylesheet" />
  <link href="app/app.component.css" rel="stylesheet" />

  <!-- 1. Load libraries -->
  <!-- IE required polyfills, in this exact order -->
  <script
src="node_modules/angular2/es6/dev/src/testing/shims_for_IE.js"></script>
  <script src="node_modules/es6-shim/es6-shim.min.js"></script>
  <script src="node_modules/systemjs/dist/system-polyfills.js"></script>
```

```

<script src="node_modules/angular2/bundles/angular2-polyfills.js"></script>
<script src="node_modules/systemjs/dist/system.src.js"></script>
<script src="node_modules/rxjs/bundles/Rx.js"></script>
<script src="node_modules/angular2/bundles/angular2.dev.js"></script>

<!-- required for http -->
<script src="node_modules/angular2/bundles/http.dev.js"></script>

<!-- required for routing -->
<script src="node_modules/angular2/bundles/router.dev.js"></script>

<!-- 2. Configure SystemJS -->
<script>
  System.config({
    packages: {
      app: {
        format: 'register',
        defaultExtension: 'js'
      }
    }
  });
  System.import('app/main')
    .then(null, console.error.bind(console));
</script>
</head>

<body>
  <pm-app>Loading App..</pm-app>

</body>

</html>

```

7. Add product-details.component.html and product-details.component.ts file under products folders.
8. Include template in product-details.component.html file.

product-details.component.html

```

<div class='panel panel-primary'>
  <div class='panel-heading'>
    {{pageTitle}}
  </div>
</div>

```

9. Add below code in your product-details.component.ts file. This will include new component called product-details.component.

product-details.component.ts

```
import {Component} from 'angular2/core';

@Component({
  templateUrl: 'app/products/product-detail.component.html'
})

export class ProductDetailComponent{
  pageTitle: string = 'Prodct Detail';
}
```

10. Do the below changes in your app.component.ts file

app.component.ts file

```
import { Component } from 'angular2/core';
import { ProductListComponent } from './products/product-list.component';
import { ProductService } from './products/product.service';
import { HTTP_PROVIDERS } from 'angular2/http';
import 'rxjs/Rx'; //Load all features
import {ROUTER_PROVIDERS, RouteConfig, ROUTER_DIRECTIVES} from 'angular2/router';
import {WelcomeComponent} from './home/welcome.component';

@Component({
  selector: 'pm-app',
  template: `
    <div>
      <nav class='navbar navbar-default'>
        <div class='container-fluid'>
          <a class='navbar-brand'>{{pageTitle}}</a>
          <ul class='nav navbar-nav'>
            <li><a [routerLink]='["Welcome"]'>Home</a></li>
            <li><a [routerLink]='["Products"]'>Product List</a></li>
          </ul>
        </div>
      </nav>

    </div>
  `,
  directives: [ ROUTER_DIRECTIVES ],
  providers: [ProductService, HTTP_PROVIDERS, ROUTER_PROVIDERS]
})

@RouteConfig([
```

```

    {path: '/welcome', name: 'Welcome', component: WelcomeComponent,
useAsDefault: true},
    {path: '/products', name: 'Products', component: ProductListComponent}
  ]
})

export class AppComponent{
  pageTitle: string= 'CAP Product Management';
}

```

11. If you look at the browser you will get menu, but it will not load appropriate view.

12. To place the view include <router-outlet> in your app.component.ts file.

app.component.ts

```

import { Component } from 'angular2/core';
import { ProductListComponent } from '../products/product-list.component';
import { ProductService } from '../products/product.service';
import { HTTP_PROVIDERS } from 'angular2/http';
import 'rxjs/Rx'; //Load all features
import { ROUTER_PROVIDERS, RouteConfig, ROUTER_DIRECTIVES } from 'angular2/router';
import { WelcomeComponent } from '../home/welcome.component';

@Component({
  selector: 'pm-app',
  template: `
    <div>
      <nav class='navbar navbar-default'>
        <div class='container-fluid'>
          <a class='navbar-brand'>{{pageTitle}}</a>
          <ul class='nav navbar-nav'>
            <li><a [routerLink]="['Welcome']">Home</a></li>
            <li><a [routerLink]="['Products']">Product List</a></li>
          </ul>
        </div>
      </nav>
      <div class='container'>
        <router-outlet></router-outlet>
      </div>
    </div>
  `,
  directives: [ ROUTER_DIRECTIVES ],
  providers: [ ProductService, HTTP_PROVIDERS, ROUTER_PROVIDERS ]
})

```



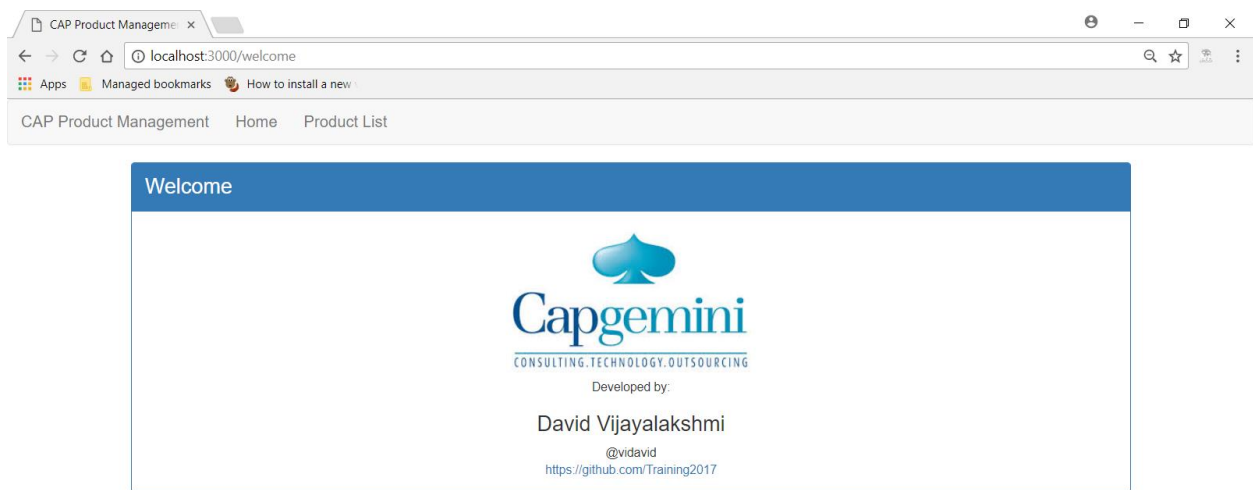
```

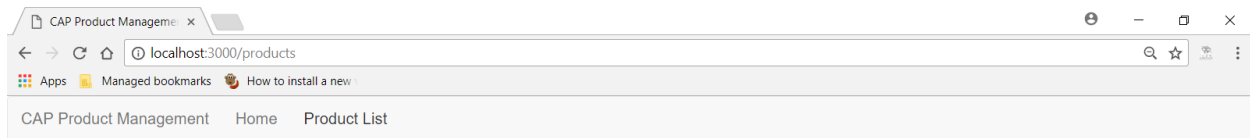
@RouteConfig([
  {path: '/welcome', name: 'Welcome', component: WelcomeComponent,
  useAsDefault: true},
  {path: '/products', name: 'Products', component: ProductListComponent}
])

export class AppComponent{
  pageTitle: string= 'CAP Product Management';
}

```

Now if you click the menu it will show the page.





Product List					
Filter by: <input type="text"/>					
Filtered by:					
Show Image	Product	Code	Available	Price	5 Star Rating
	Leaf Rake	gdn-0011	March 19, 2016	\$19.95	★★★★
	Garden Cart	gdn-0023	March 18, 2016	\$32.99	★★★★
	Hammer	tbx-0048	May 21, 2016	\$8.90	★★★★★
	Saw	tbx-0022	May 15, 2016	\$11.55	★★★★
	Video Game Controller	gmg-0042	October 15, 2015	\$35.95	★★★★★

localhost:3000/products

Passing Parameters to Router:

13. Add one more path under @RouteConfig in your app.component.ts file.

app.component.ts

```
import { Component } from 'angular2/core';
import { ProductListComponent } from './products/product-list.component';
import { ProductService } from './products/product.service';
import { HTTP_PROVIDERS } from 'angular2/http';
import 'rxjs/Rx'; //Load all features
import { ROUTER_PROVIDERS, RouteConfig, ROUTER_DIRECTIVES } from 'angular2/router';
import { WelcomeComponent } from './home/welcome.component';
import { ProductDetailComponent } from './products/product-detail.component';

@Component({
  selector: 'pm-app',
  template: `
    <div>
      <nav class='navbar navbar-default'>
        <div class='container-fluid'>
          <a class='navbar-brand'>{{pageTitle}}</a>
          <ul class='nav navbar-nav'>
            <li><a [routerLink]="['Welcome']">Home</a></li>
            <li><a [routerLink]="['Products']">Product List</a></li>

          </ul>
        </div>
      </nav>
    </div>
  `
})
```

```

        </div>
    </nav>
    <div class='container'>
        <router-outlet></router-outlet>
    </div>
</div>
`,
    directives: [ ROUTER_DIRECTIVES ],
    providers: [ ProductService, HTTP_PROVIDERS, ROUTER_PROVIDERS ]
})

@RouteConfig([
    {path: '/welcome', name: 'Welcome', component: WelcomeComponent,
    useAsDefault: true},
    {path: '/products', name: 'Products', component: ProductListComponent},
    {path: '/product/:id', name: 'ProductDetails', component:
    ProductDetailComponent}
])

export class AppComponent{
    pageTitle: string= 'CAP Product Management';
}

```

14. Modify your product-list.component.ts file

product-list.component.ts

```

<div class='panel panel-primary'>
    <div class='panel-heading'>
        {{pageTitle}}
    </div>
    <div class='panel-body'>
        <div class='row'>
            <div class='col-md-2'>Filter by:</div>
            <div class='col-md-4'>
                <input type='text' [(ngModel)]='listFilter'/>
            </div>
        </div>
        <div class='row'>
            <div class='col-md-6'>
                <h3>Filtered by: {{listFilter}} </h3>
            </div>
        </div>
        <div class='table-responsive'>
            <table class='table' *ngIf='products && products.length'>

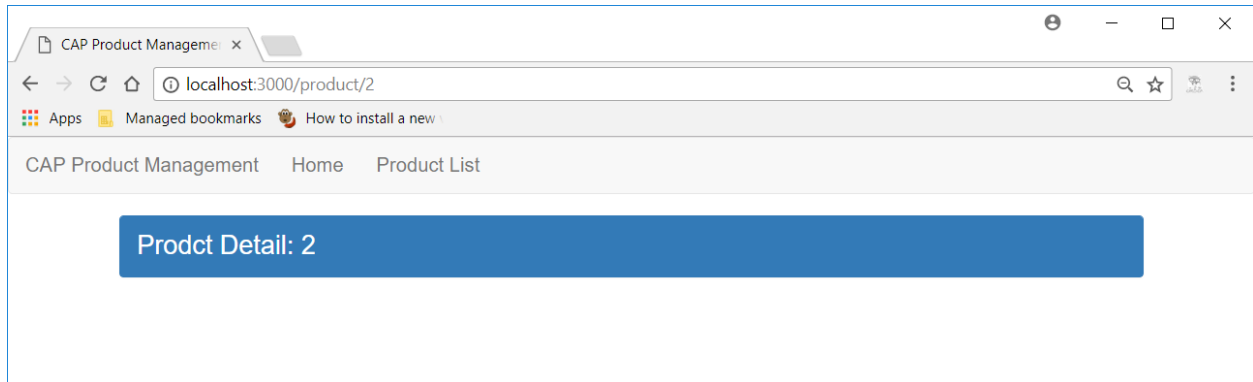
```

```

        <thead>
            <tr>
                <th>
                    <button class='btn btn-primary'
                        (click) = 'toggleImage()'>
                        {{showImage ? 'Hide' : 'Show'}} Image
                    </button>
                </th>
                <th>Product</th>
                <th>Code</th>
                <th>Available</th>
                <th>Price</th>
                <th>5 Star Rating</th>
            </tr>
        </thead>
        <tbody>
            <tr *ngFor='#product of products | productFilter:listFilter'>
                <td>
                    <img *ngIf='showImage'
                        [src]='product.imageUrl'
                        [title]='product.productName'
                        [style.width.px]='imageWidth'
                        [style.margin.px]='imageMargin'>
                    </td>
                    <td>
                        <a [routerLink]="['ProductDetail',{id: product.productId }]">
                            {{product.productName}}
                        </a>
                    </td>
                    <td>{{product.productCode | lowercase }}</td>
                    <td>{{product.releaseDate}}</td>
                    <td>{{product.price | currency:'USD':true:'1.2-2'}}</td>
                    <td>
                        <ai-star [rating]='product.starRating'
                            (ratingClicked)='onRatingClicked($event)'></ai-
star>
                    </td>
                </tr>
            </tbody>
        </table>
    </div>
</div>
</div>

```

When you click productname you can see below output:



Activate route with code:

15. Now we are going to add back button in the product-details.component page.
16. Add below code in your product-details.component.ts file.

product-details.component.ts

```
import {Component} from 'angular2/core';
import {RouteParams, Router} from 'angular2/router';

@Component({
  templateUrl: 'app/products/product-detail.component.html'
})

export class ProductDetailComponent{
  pageTitle: string = 'Prodct Detail';

  constructor(private _routeParams: RouteParams,
               private _router: Router){
    let id = +this._routeParams.get('id');
    this.pageTitle += `: ${id}`;
  }

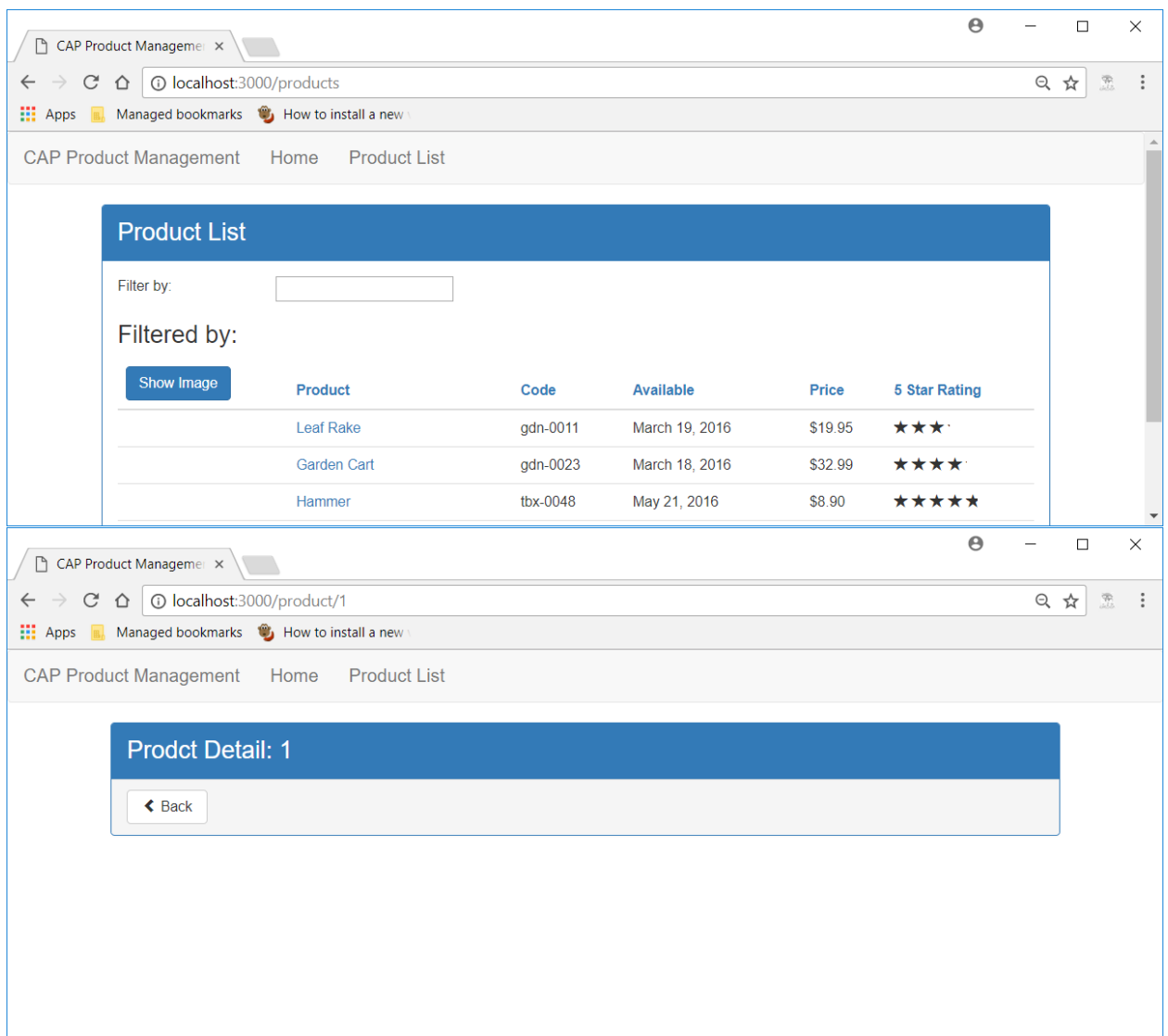
  onBack(): void{
    this._router.navigate(['Products']);
  }
}
```

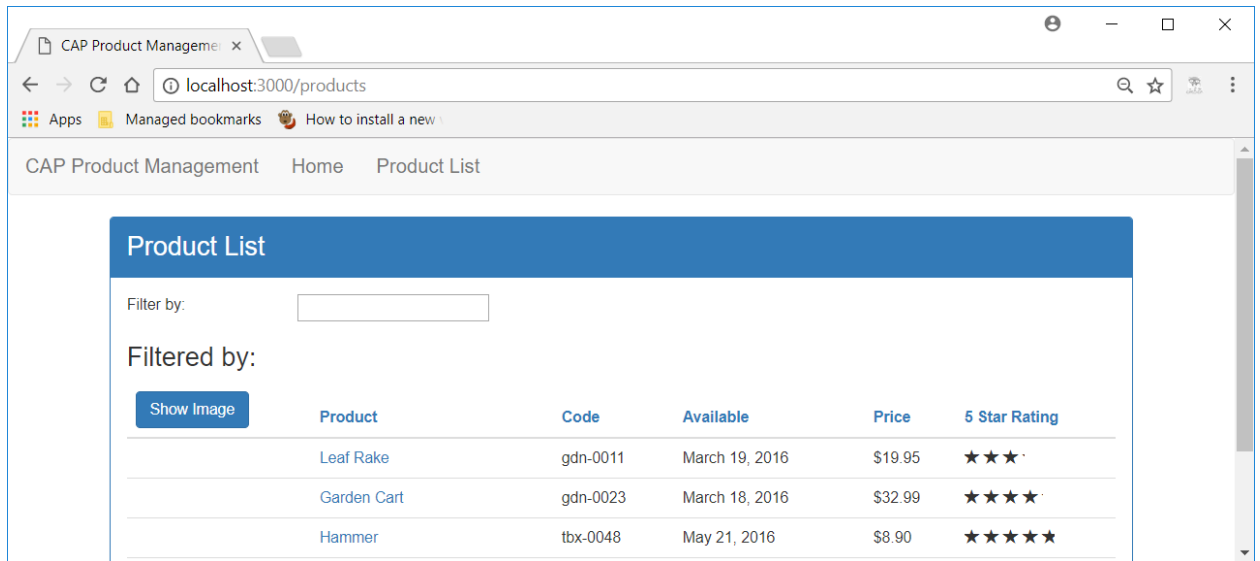
17. Now include below code in product-details.component.html file.

product-details.component.html

```
<div class='panel panel-primary'>
  <div class='panel-heading'>
    {{pageTitle}}
  </div>
  <div class='panel-footer'>
    <a class='btn btn-default' (click)='onBack()' style='width:80px'>
      <i class='glyphicon glyphicon-chevron-left'></i> Back
    </a>
  </div>
</div>
```

Now you can click any one of the productname in your productlist page it will show product details page with back button, and it works now.





Angular CLI:

```
npm install -g @angular-cli
```

```
ng new my-app
```

```
cd my-app
```

```
ng serve --open
```